

**EV316935517**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

**APPLICATION FOR LETTERS PATENT**

**Multi-Axis Television Navigation**

Inventors:

Samuel Thomas Scott, III

David Hendler Sloo

Peter T. Barrett

**ATTORNEY'S DOCKET NO. MS1-1731US**

## **COMPUTER PROGRAM LISTING**

One .XSD text file listing and one .XML text file listing used in accordance with the subject matter are provided in an appendix after the abstract on 3 sheets of paper and incorporated by reference into the specification. The .XSD text file is an exemplary sample extensible markup language schema definition file for a metadata database and the .XML text file is an exemplary sample database suitable for multi-axis TV navigation.

## **TECHNICAL FIELD**

The subject matter relates generally to multimedia data communications and specifically to multi-axis television information grid navigation.

## **BACKGROUND**

Conventionally, television programming and its metadata guide information has been organized according to original broadcasting schedules into channels, characterized by a given program on a given channel at a given time. End users and broadcasters alike have become accustomed to the ingrained concept of navigating for programs in terms of a two-dimensional program listings grid, upon which for a given time point on a time axis of the grid each channel on a channel axis of the grid has a different program, reflected in language, such as "my favorite show will be on channel six at 8:00 pm." The time/channel information grid, however, is an arbitrary technique for defining a user's desire to find out what is on television.

A television channel represents a specific sequence of programs over time. Broadcasters conventionally determine the programming that they will carry

1 during a future time interval, making the two axes of channel and time a natural  
2 and convenient way to organize television programming content and television  
3 programming metadata, that is, the guide information sometimes referred to as  
4 electronic program guide (EPG) data.

5 **Fig. 1** shows metadata 100 for a program comprising multimedia content to  
6 be shown on television as a “TV show.” Of many attributes included in the  
7 metadata 100, only two attributes, namely channel 106 and time 108, are utilized  
8 to place an identifier of the program, such as the title 102, into a conventional  
9 program listings grid (or guide) 104.

10 **Fig. 2** shows the conventional program listings grid 104 of Fig. 1 in greater  
11 detail. The channel axis 202 is plotted against the time axis 204, as above, and  
12 provides the context for navigating among television programs.

13 At a given time point 206, usually selected to be the present moment,  
14 navigation conventionally consists of traversing from program to program along  
15 the channel axis 202. Most conventional TV systems have been designed to  
16 navigate according to a channel axis model, in which navigation of both programs  
17 and their respective program guide information typically follows the conventional  
18 channel axis 202 versus time axis 204 format. That is, as shown in **Fig. 3**, a  
19 navigation control, such as a “channel changer” located either on a TV set or on a  
20 remote control, traverses the channel axis 202 via, for example, “channel change  
21 keys 300,” such as a “next” key 302 and a “previous” key 304. An “enter” or  
22 “select” key 306 often accompanies channel change keys 300. Some navigation  
23 controls that use channel change keys 300 also include left and right keys that may  
24 have a “first” and “last” function or perhaps a volume control function. The  
25

1 illustrated channel change keys 300 are presented, of course, only as one example  
2 of conventional navigation controls.

3 In traversing a channel axis 202, conventional navigation techniques  
4 generally hold the time axis 204 constant at a single point representing the present,  
5 and cycle through all available channels, displaying the program that is currently  
6 playing on each channel in turn, i.e., as a user changes channels the program  
7 currently playing on a particular channel automatically displays. The two  
8 attributes, channel and time, provide the context for conventional television  
9 programming navigation. Navigating from one program or program reference  
10 (program ID) to another is sometimes referred to as "scrolling" or "channel-  
11 surfing." Each program ID may be used to access stored data in a local, remote, or  
12 DVR media store.

13 Delivery of programs in a sequence over channels is the best-known  
14 context for organizing and navigating television content and characterizes a  
15 lingering and predominant mindset in the multimedia arts. However, there is now  
16 too much multimedia content to rely on conventional navigation techniques. It is  
17 awkward to navigate hundreds or even thousands of conventional channels.

18 Digital storage of television programs with its characteristic random access  
19 feature frees navigation systems from accessing only programs that are being  
20 broadcast in the present: program access is no longer limited to currently  
21 broadcasted content. Accordingly, the amount of content to choose from  
22 potentially consists of programs from thousands of channels (several hundred  
23 active channels in some present systems) multiplied by the content of each channel  
24 over a time period, which can be years. Although the system of navigating  
25 programs by a channel axis 202 and a time axis 204 persists, there is far too much

1 multimedia content to be efficiently and enjoyably navigated by such a system.  
2 As Digital Video Recording (DVR) technology evolves, vast stores of content are  
3 becoming available to users as “everything on demand,” making ease of  
4 navigation a priority.

## 5 6 **SUMMARY**

7 An exemplary multi-axis television navigation system defines television  
8 navigation axes according to attributes of television programs. In one  
9 implementation, if a television program has an attribute defining an axis, then the  
10 exemplary system links a predefined database query associated with the axis to  
11 metadata for the television program. When launched, the predefined query returns  
12 a sequence of navigable television programs having the attribute that defines the  
13 axis.

## 14 15 **BRIEF DESCRIPTION OF THE DRAWINGS**

16 Fig. 1 is a graphic representation of a conventional television program  
17 guide information.

18 Fig. 2 is a graphic representation of a conventional program listings guide.

19 Fig. 3 is a graphic representation of conventional channel change  
20 navigation keys.

21 Fig. 4 is a graphic representation of an exemplary navigational axis based  
22 on an episode attribute.

23 Fig. 5 is a graphic representation of an exemplary navigational axis based  
24 on an actress attribute.  
25

1        Fig. 6 is a graphic representation of a multiple navigational axes combined  
2 in a two-dimensional grid.

3        Fig. 7 is a graphic representation of various types of navigational axes  
4 based on program attributes.

5        Fig. 8 is a graphic representation of multi-axis television navigation.

6        Fig. 9 is a block diagram of an exemplary multi-axis television navigation  
7 system.

8        Fig. 10 is a block diagram of an exemplary multi-axis television navigation  
9 engine.

10       Fig. 11 is a block diagram of an exemplary predefined query structure.

11       Fig. 12 is a block diagram of another exemplary predefined query structure.

12       Fig. 13 is a graphic representation of an exemplary tree user interface for  
13 presenting axes for selection.

14       Fig. 14 is a graphic representation of an exemplary information page user  
15 interface for presenting axes for selection.

16       Fig. 15 is a flow diagram of an exemplary method of multi-axis television  
17 navigation.

## **DETAILED DESCRIPTION**

### **Overview**

The subject matter describes the use of database technology and digital storage to establish new axes, or logical sequences of programs for television exploration (“navigation”) in addition to conventional axes known as broadcast channels. When television content is stored digitally, contexts for content discovery and navigation are not limited to conventional chronological sequences, indeed each program with its related attributes can stand on its own, and programs can be combined into navigable sequences based on other contexts besides chronological sequence. Conventional time/channel information grids generated by chronological sequences are an arbitrary technique for defining a user’s desire to find out what is on television. The multiple combinable axes described herein allow the user to define custom navigational views that deliver the most useful information suited to the particular user, tailored to the user’s taste.

The subject matter enables navigation on arbitrary axes without having to initiate explicit search commands. Predefined axes, such as “actor” and “episode” may be combined with Boolean operators to yield even more navigation contexts.

A navigational axis (“axis”) comprises a sequence of programs or more precisely, program identifiers (IDs) that are included in the sequence for some logical reason, usually a common attribute. Conventionally, a channel is one type of axis wherein programs are included in the axis sequence because they all proceed from a common broadcast point. Sequential ordering of programs within an axis may be left random or an ordered sequence may be imposed for an additional logical reason, for example, a conventional channel axis 202 has a particular temporal order that reflects the time that the included programs were

1 broadcast. **Fig. 4** shows an exemplary axis 400 configured by episode sequence,  
2 allowing a user to navigate among programs represented on the episode axis. In  
3 this example, the user watching the October 6th episode would be able to navigate  
4 to the September 29th episode (e.g., a “previous” key) or the October 13th episode  
5 (e.g., a “next” key).

6 “Navigation,” as presented herein with respect to exemplary axes, refers to  
7 changing television channels to view different programs. Although with respect to  
8 exemplary systems and methods described herein “navigation” can also be applied  
9 to browsing just television programming information, i.e., EPG metadata, many  
10 implementations of the exemplary systems and methods describe navigating  
11 program content, not just program guide information. Hence, for example, when a  
12 user selects an exemplary “actor” axis specifying the actor Russell Crowe,  
13 navigation controls on the user’s equipment change their function from cycling  
14 through programs based on broadcast channels to cycling between programs in  
15 which Russell Crowe plays a starring role. **Fig. 5** shows another exemplary  
16 navigation axis 500 by actor name. In this example, a user can navigate to  
17 “Before and After” 502 (e.g., “previous” key) and “Heartburn” 504 (e.g., “next”  
18 key) with a single click.

19 Exemplary methods also allow logical combination of several exemplary  
20 axes to achieve even more refined or more extensive navigational contexts, for  
21 example, an episode axis specifying “M.A.S.H.” and a year axis specifying “1978”  
22 result in a different episode of M.A.S.H. from 1978 playing on the user’s  
23 equipment each time the user actuates “channel change” controls. **Fig. 6** shows  
24 another type of exemplary axis combination, in which navigation of two discrete  
25 axes is enabled via a two-dimensional grid 600. In this example, analogous to the

1 time/channel conventional program guide grid 104, a first column 602 represents  
2 an axis of different actors, while rows appended to the first column 602 each  
3 represent movies of one respective actor. A user can navigate by different actors  
4 and by movies of different actors based on the combination of axes.

5 In one implementation of exemplary navigation, when a user changes  
6 programs on an axis, e.g., presses a channel change key to interrupt one episode of  
7 GILLIGAN'S ISLAND to view another episode of the same show, the interrupted  
8 episode resumes where it left off when the user returns. This allows a user to  
9 compare movies and episodes and to jump in and out of programs without missing  
10 any content.

11 Practically speaking, in most implementations, an exemplary navigational  
12 system might provide only four or five additional predefined axes (that can be  
13 further combined with each other) besides the well-known conventional channel  
14 and time axes, but many more axes than four or five are possible in an exemplary  
15 system since almost unlimited attributes can be used to describe a program.

16 When a user combines two or more navigational axes to yield a more  
17 refined navigational context, for example, axes combined to set up a navigational  
18 context of "FRIENDS" episodes AND "Elliot Gould" as actor AND "1997" as the  
19 relevant year, the user is relieved of the burden of finding a likeable program  
20 amidst large quantities of content. Rather than having to type in three search  
21 terms "FRIENDS," "Elliot Gould," and "1997" and supply the Boolean operator  
22 "AND" to launch a conventional database search of a content repository, instead  
23 links to predetermined queries that compute the axes are supplied in context: links  
24 to predefined navigational axes are embedded within overlays, content, and/or  
25 metadata being currently displayed in ways that are logically meaningful, i.e., "in

1 context” as the user navigates. Hence, with a single selection or “click” a user can  
2 extend a context attribute from a present program, e.g., “actress Meryl Streep,”  
3 into an axis having a navigational context of other Meryl Streep programs  
4 accessible one after another through the navigation controls that selected the  
5 Meryl Streep context attribute. Data associations for fulfilling the predetermined  
6 queries can be pre-rendered on a server or in a client. Pre-rendering on a client  
7 can offer its own processing advantages.

8 Exemplary axes are made available to a user as a link, icon, menu tree, etc.,  
9 for example, in a UI that can be called up to show program metadata, including the  
10 links or icons to select axes. Alternatively, axes may be selected using cues, links,  
11 controls, etc., that are only on a remote control so that program metadata is never  
12 seen while a program associated with the metadata is playing on a display. For  
13 example, after pressing a “select” key 306 as a second function key, actuation of  
14 the right arrow key might always select an “actor” navigational axis.

15 In one implementation, axes that are likely to be selected in the context of a  
16 TV program that is currently playing are highlighted (e.g., by default) in program  
17 data or overlays accessible through the select key 306, etc., that allow axis  
18 switching. The user selects among axes and combinations of axes offered, but  
19 does not have to create new axes from scratch. In other words, as noted above,  
20 axis choices can be laid out for selection in the context of a show that is currently  
21 playing and selection of one or more axes launches pre-determined queries.

22 For comparison, some features of the subject matter render exemplary  
23 program navigation on exemplary axes like surfing an Internet web, where  
24 hyperlinks are pre-placed in context in displayed content and new content revealed  
25 by clicking a hyperlink is pre-rendered. Thus, a user of an exemplary navigation

1 system can move through content in a finely tuned logical manner using only a TV  
2 remote control or other navigation tool that has minimal navigation controls. The  
3 subject matter provides the added benefit that each time an exemplary axis and/or  
4 context is selected, navigation keys on the user's navigation tool, e.g., a TV  
5 remote controller, are converted to "changing channels" according to the newly  
6 selected axis and/or context until axes are changed again.

7 The subject matter takes advantage of the fact that often a user's current  
8 program or position within the program guide information provides a great deal of  
9 context and can act as a starting point or branching point for further navigation.  
10 Thus, if a user is currently focused on a science fiction movie, then an overlay for  
11 the science fiction movie providing opportunities to switch axes can include axis  
12 selections that represent logical "next steps" in navigation relative to the science  
13 fiction movie: for instance, more science fiction movies, more movies by the  
14 same director, more movies with the same actor, etc.

15 This detailed description describes subject matter that includes methods,  
16 engines, database structures, and query techniques for creating many types of  
17 navigational axes according to attributes of programs and their contexts in addition  
18 to conventional channel and broadcast-time attributes.

### 19 20 **Multiple Navigational Axes**

21 **Fig. 7** shows program metadata 100 for a movie, "The Mountain," available  
22 as a program on television. The metadata 100 includes many attributes and  
23 respective values for some of the attributes. Some attributes have an "either/or" or  
24 digital character (such as, "high definition TV" (HDTV) (y/n?)) while other  
25 attributes can assume a variable value within a known set of values, such as an

1 actress name for an actress attribute, or the year 1978 within a range of years.  
2 Attributes may include the type of program (movie, TV series, etc.), program title,  
3 alphabetical order of title, year of release, channel, time, first air date, episode  
4 order, episode name, genre, actors, writer, director, producer, rating, sound  
5 characteristics, video characteristics, language, subtitles, closeness of match to  
6 search criteria, popularity, and many other attributes. Programs can have almost  
7 unlimited attributes, each of which can be used as a basis for configuring a  
8 navigational axis.

9 When a “director” attribute 702 is available as the context for a “director”  
10 navigational axis 704, an instance of the director axis 704 can be configured by  
11 also selecting the name “Edward Dmytryk” 706 as a value for the director attribute  
12 702. A sequence of program IDs, if programs for such exist in an operative  
13 database, can be computed wherein each program ID points to a program that has  
14 the value. The director axis 704 can then be navigated by using navigational  
15 controls, such as channel change keys (e.g., 302, 304 and possibly 306), on a  
16 user’s equipment, such as a TV set, television enabled computer, or TV remote  
17 control.

18 An exemplary application or multi-axis TV navigation engine (e.g., 904  
19 below) changes the functioning of navigation controls (e.g., 300) substituting a  
20 new axis for the conventional channel axis 202 usually navigated by the  
21 navigation controls (e.g., 300). When a user presses one of the channel change  
22 keys 300, for instance, various events can occur depending on the implementation.  
23 In one implementation, a television navigation system is concerned mainly (or  
24 only) with the program guide information. In this case, navigating a “director”  
25

1 axis 704 results in the user browsing only the program guide information for each  
2 program ID associated with the selected director in the director axis 704.

3 In another implementation, an exemplary television navigation system  
4 displays actual programs for each program ID in an axis. Thus, pressing a channel  
5 change key (e.g., 302) changes between programs directed by the person named in  
6 the attribute value. If the attribute upon which an axis is based is “episode” then  
7 actuating a channel change key 302 would change between episodes of a subject  
8 program and display each different episode with each press of a channel change  
9 key 302.

10 Mechanisms for selecting axes to navigate and selecting values for the axes  
11 will be discussed more fully below with respect to Figs 13 and 14.

12 **Fig. 8** shows various exemplary navigational axes being navigated in turn  
13 by the same navigation controls, such as a set of channel change keys 300. Of  
14 course, channel change keys 300 are only one example of navigational controls for  
15 multi-axis television navigation. Different operational controls could be used,  
16 such as a dial, a touch pad, a keyboard, a mouse, a wheel, trackball, displayed  
17 rolling cylinder, onscreen keys, onscreen grid, onscreen cascade of cards, order  
18 forms, voice activated navigator, etc. Channel change keys 300, however,  
19 demonstrate that exemplary multi-axis navigation can be accomplished via a  
20 minimum of controls, such as two or three buttons or keys.

21 In Fig. 8, a user is currently viewing a program 800, the “Mountain.” In  
22 one implementation, the user has selected the program 800 by refraining from  
23 further channel changes, leaving the program 800 playing on a display. The user  
24 may then decide to change navigational axes.

1 A change of navigational axes may be implemented in various ways. In  
2 one implementation an actuation of a “select” key 306 calls up a menu overlaying  
3 the program and converts the “previous” and “next” keys 302, 304 to movement  
4 keys for navigating the menu until the select key 306 is actuated again, for  
5 example, to select a different axis. The menu may provide a list of axes to choose  
6 from. In another implementation, a press of a select key 306 allows other keys  
7 302, 304 to cycle choices on a list of axes but a menu is not displayed on the same  
8 display as the program, however, the list of axes may be displayed on a remote  
9 controller bearing the navigation controls, such as channel change keys 300. In  
10 yet another implementation, when only program information (metadata 100)  
11 corresponding to a program ID is displayed instead of the program itself, then the  
12 metadata 100 may have embedded hyperlinks or icons that allow a user to select  
13 axes and/or values for the axes.

14 Regardless of which implementation of axis selection is employed, the user  
15 now changes navigational axes to an “actor” axis 802 that has a “Spencer Tracy”  
16 value for the actor attribute and a “movie” axis filter. Some attributes have only  
17 one possible value, such as the year of movie release, so that in one application a  
18 value for an axis choice defaults to a single value supplied by the metadata 100  
19 without any user intervention. An exemplary application may also default to a  
20 first name on a list in the metadata 100, etc. Scrolling while the “Spencer Tracy as  
21 actor” axis 802 is selected results in a cycling through available Spencer Tracy  
22 movies, each of which are played in turn as the user scrolls. If the user stops on  
23 (selects) the movie “The Old Man and the Sea” 804, which starred Spencer Tracy,  
24 then “The Old Man and the Sea” 804 becomes a starting point if navigational axes  
25 are again switched.

1       The user switches to a “writer” axis 806. The writer of “The Old Man and  
2 the Sea” 804 was Ernest Hemingway, which is inserted as a value in a writer  
3 attribute upon which the writer axis 806 is at least in part configured. Now when  
4 the user scrolls, available movies based on books written by Ernest Hemingway  
5 are played in turn on the television display as the user scrolls with the channel  
6 change keys 300. The user pauses on the movie “The Sun Also Rises” 808, which  
7 becomes the next starting point for navigation.

8       The user switches to a “date” axis 810. A value of “1984” is supplied by  
9 default as the year of release of the movie “The Sun Also Rises” 808, which was  
10 the context or starting point from which axes were changed. As the user scrolls,  
11 movies released in 1984 are played in turn on the TV display. The user stops on  
12 or otherwise selects the movie, “Mad Max” 812, which was released in 1984.

13       The navigational path through television programming content can continue  
14 indefinitely with any number of axis changes. A user might continue by  
15 reselecting an actor axis 814 and scrolling to arrive at the movie, “Braveheart”  
16 816. From there, a genre axis 818 with the value “war & combat” could be  
17 selected and scrolling might arrive at the movie, “Gladiator” 820. Selection of a  
18 “director” axis 822 with a value of “Ridley Scott” from “Gladiator” 820 and  
19 scrolling might further arrive at the movie, “Blade Runner” 824, and so on.

### 20 21       **Exemplary Multi-Axis TV Navigation System**

22       Fig. 9 shows an exemplary multi-axis TV navigation system 900. The  
23 illustrated example is only one configuration of an environment suitable for  
24 practicing the subject matter, many other configurations are possible.

1 A server 902, which may serve as a hub for various connected multimedia  
2 devices, includes an exemplary multi-axis TV navigation engine (“navigation  
3 engine”) 904. A headend 906, a node 908, such as a set-top box, a digital video  
4 recorder (DVR) 910, a television 912, and a remote controller 914 are also  
5 communicatively coupled as illustrated.

6 The remote controller 914 includes a navigation controller, such as  
7 “channel change keys” 300, which provide an agency for selection of program  
8 content, program metadata 100, and navigational axes via the multi-axis TV  
9 navigation engine 904. Although the navigation engine 904 is shown as included  
10 only in the server 902, in alternate implementations, various components of a  
11 navigation engine 904 could be located in different parts of the system 900. When  
12 the navigation engine 904 changes, computes, or creates an axis, the axis consists  
13 of a sequence of program IDs. Program content corresponding to each program  
14 ID may be accessed for play from the server 902, the headend 906, the DVR 910,  
15 etc. Additionally, if a particular system is set up to display program metadata 100  
16 corresponding to program IDs on an axis, the program metadata 100 may also be  
17 accessed from a node 908 in which EPG data, for example, can be stored, or from  
18 the navigation engine 904 itself, if it includes a metadata cache.

### 19 20 **Exemplary Multi-Axis TV Navigation Engine and Data Structure**

21 **Fig. 10** shows an exemplary multi-axis TV navigation engine 904, in  
22 greater detail than Fig. 9. An exemplary navigation engine 904 may include a  
23 metadata cache 1002, a multi-axis database schema 1004, and a metadata database  
24 1006, which may be broken out into multiple attribute indices 1008. A user  
25 interface (UI) 1010 may include an axis selector 1012 that includes a combiner

1 1014, and a navigation controller 1016, or in some implementations, a navigation  
2 controller interface (not shown). The user interface 1010 is coupled to  
3 communicate with a query engine 1018 via one or more predefined queries 1020.  
4 The query engine is coupled with an axis cache 1022 that includes one or more  
5 lists of program identifiers. The axis cache 1022 is communicatively coupled with  
6 a content access engine 1026. The illustrated engine can be implemented in  
7 hardware and/or software, and is merely one example of a navigation engine 904  
8 suitable for practicing multi-axis TV navigation.

9 In operation, the metadata cache 1002 receives an input of EPG metadata  
10 100 from a headend 906 of a content provider. The metadata 100 may be in  
11 various formats, depending on its source. The multi-axis database schema 1004  
12 arranges, organizes, and/or parses the metadata 100 into a metadata database for  
13 compatibility with the query engine 1018 and the predefined queries 1020.

14 Since many different kinds of databases can be used for the metadata  
15 database 1006, the data structure adopted depends on the type. In relational or  
16 hierarchical database systems, the multi-axis database schema may index program  
17 attributes and/or certain values for the program attributes into index tables or child  
18 databases. In an extensible markup language (XML, etc.) system, a multi-axis  
19 database schema 1004, such as that represented by the extensible markup language  
20 schema definition (XSD) shown in Appendix A, entitled: "Sample Exemplary  
21 Schema To Organize EPG Metadata To Support Multi-Axis TV Navigation,"  
22 specifies a self-referential data structure suitable for multi-axis TV navigation. An  
23 XML example of a self-referential metadata database 1004 is shown in Appendix  
24 B, entitled "Sample XML Database for Multi-Axis TV Navigation." The sample  
25

1 metadata database shown in Appendix B includes four television programs and  
2 two axes and demonstrates one manner of expressing the definition of an axis.

3 To recapitulate, a database schema 1004 defines the data representation  
4 model: the database schema 1004 organizes metadata 100 in a way that supports  
5 declarative queries to generate axes, that is, the sequences of programs or program  
6 IDs. From a data representation model, axes may be built, e.g., using indices and  
7 predefined queries, on a server 902 and made available for the UI 1010 as  
8 iterators. A user can set up a context, e.g., movies with Russell Crowe as an actor,  
9 and then iterate or navigate, e.g., “next,” “previous,” “first,” “last,” etc. (“first” can  
10 show a first episode in a series and “last” shows the last episode), over programs  
11 in the constructed axis.

12 An exemplary database schema 1004 captures program metadata 100  
13 including, e.g., program title information and role information as it is received  
14 from a content provider and builds discrete or integrated indices and/or tables  
15 representing axes selected to be in the particular exemplary navigational system  
16 900.

17 In one implementation, a database schema 1004 is in a global listings  
18 format (GLF), using a GLF relational model for television programming metadata  
19 100 as described in U.S. Patent Application No. 10/356,694 to Andrew Simms and  
20 Samuel Thomas Scott, III, entitled: “Global Listings Format (GLF) for  
21 Multimedia Programming Content and Electronic Program Guide (EPG)  
22 Information.” Television metadata 100 that is received in a GLF (format), which  
23 has self-referential structure and contains interlocking and cross-locking lineup,  
24 program, and scheduling data is already imbued with a great deal of relational  
25 character. GLF data is reliable, because during creation the interlocking and cross-

1 locking fields are mandatory and may be screened for accuracy. In a GLF schema,  
2 moreover, relationships between related pieces of information are built into the  
3 fabric of the global listings format.

4 Whether or not a GLF is used for metadata 100 or the metadata database  
5 1006, it should be noted that selection of a database schema 1004 may depend on  
6 the anticipated complexity of an exemplary navigation system 900. In one  
7 implementation, each service provider may have different axis index definitions,  
8 or in another implementation all axis index definitions are uniform for all  
9 participating content providers. For example, a premium service group has more  
10 complex content that may be assigned many attributes (e.g., users navigating over  
11 HOME BOX OFFICE, the SOPRANOS, R-rated movies, etc.) while a basic  
12 service group might be utilizing a different data center with lightweight content,  
13 wherein programs are assigned only a few attributes, such as release year and  
14 actors' names. Thus, when a subscriber upgrades from a lower service to a  
15 premium service more axes may be made available by the content provider.

16 In one implementation, one or more axis specifications process EPG  
17 metadata 100 into indices 1008 set up to assist predefined queries, which construct  
18 the axes. In this type of exemplary system, when a user selects an axis in the  
19 context of a default or selected attribute, for example an attribute selected from  
20 program currently being displayed, a predefined query associated with the selected  
21 axis runs through one or more appropriate indices 1008 to construct the selected  
22 axis, i.e., the list of program IDs. Depending on the format used for the EPG  
23 metadata 100, the formation of workable indices 1008 may require iterating and  
24 sorting out the newly revised EPG metadata 100 for each program, one program at  
25 a time. Program axis queries against a database 1006 then describe relationships

1 between the indices 1008, and allow the database 1006 to produce the desired  
2 query response, thereby constructing a resulting axis. Thus, a query to the server  
3 902 encapsulates commands to produce content on a new axis. Query  
4 composition can span one or more axes. A simple axis represents a single  
5 program attribute, whereas a complex axis represents more than one attribute. For  
6 example, a simple axis might represents "actor's name" specifying "Smith," while  
7 a complex axis might represent "actor's name" specifying "Smith" and "four star  
8 rating." Perhaps the user has just purchased a high definition TV and only wants  
9 to find HDTV programs amidst the content. If the navigation system 900 allows,  
10 the user can add a HDTV axis to specify only HDTV content.

11 In the exemplary UI 1010, the user may employ the navigation controller  
12 1016 to call up an axis selector 1012, e.g., actuate a select key 306 to produce an  
13 onscreen overlay of program metadata 100 having embedded axis links. The  
14 navigation controller 1016 may then allow selection of one or more links. In one  
15 implementation, each axis selected has a predefined query 1020. Multiple axes  
16 may be joined by logical operators via the combiner 1014. For example, Boolean  
17 operators, such as AND, OR, XOR, NOT, etc. may be inserted between query  
18 terms to produce combination axes. Filtering logic may also be used to hone an  
19 axis to a very narrow set of program IDs.

20 The query engine 1018 may be a database engine created especially for the  
21 task of optimized metadata database 1006 queries, but in most implementations,  
22 standard database engines and tools, such as those included in MICROSOFT®  
23 SQL SERVER 2000 DATABASE SERVER work well in an exemplary  
24 navigation engine 904 (Microsoft Corp., Redmond, Wa). The query engine 1008  
25 returns an axis, that is, one or more lists of program identifiers 1024, to an axis

1 cache 1022. Then, a program content access engine 1026, as directed by the  
2 navigation controller 1016, reads a program ID from the list of program identifiers  
3 1024 and retrieves a television program for display. The program content access  
4 engine 1016 may be set up to retrieve content from multiple sources, such as a  
5 remote headend 906 or storage media, local storage media on a server 902, a  
6 networked DVR 910, etc.

7 **Fig. 11** shows an exemplary query structure 1100, illustrating the key-value  
8 pair associations with program data. Using a relationship between the key-value  
9 pair associations with program data, the query establishes and produces an axis of  
10 program data for navigating by an episode attribute of program descriptions.

11 **Fig. 12** shows a query structure 1200, illustrating an actor-based axis  
12 construction. Using relationships between the key-value pair associations and the  
13 program data, the query establishes an axis of program data for navigating by the  
14 actor name attribute of program descriptions.

### 15 **Exemplary User Interfaces**

16  
17 Exemplary UIs 1010 offer ways to select axes, that is, ways to select among  
18 data sets generated by the predefined queries 1020. The selection of multiple axes  
19 may be made available in the context of a UI 1010, as noted above, whether the UI  
20 1010 is a grid, a program information page, icons, video thumbnails, etc. For  
21 example, in one exemplary UI 1010, clicking on a link or icon for one axis may  
22 display a hint, such as a highlight or dotted line outlining a cue word, that an  
23 alternate and/or additional axis is available to be iterated over. Or again, clicking  
24 on the word “page” on a displayed page of EPG metadata 100 might cause a cue  
25 word or icon for a movie axis to be highlighted. Perhaps another click highlights

actors, building a combination of ever more refined navigation contexts using additional axes, with each click.

**Fig. 13** shows part of an exemplary program attribute tree 1300 that can be used in various ways as an exemplary UI 1010 or as part of an exemplary UI 1010. In one implementation, the tree is displayed only on a remote controller 914 with navigation controls 1016 for highlighting and selecting attributes corresponding to axes. For many navigation systems 900, the exemplary tree 1300 only needs to include a few member attributes. In another implementation, the exemplary tree 1300 or parts thereof appears onscreen on the same display as the program content, for example, as a menu overlay. There are many types of UIs 1010, however, besides the exemplary tree 1300 that can be used for displaying axes for selection in an exemplary navigation system 900.

**Fig. 14** shows an exemplary UI 1010 comprising an onscreen information page 1402 for each program ID on an axis. When a user navigates over the axis, which is associated with a name value for an actress, the respective information page 1402 for each current program ID comes to the front. The actress name (in the solid selection box) 1404 and movie attribute (in the dotted selection box) 1406 indicate that both attributes (name, movie) will be use in a Boolean “AND” fashion to determine navigation behavior. A video window 1408 may be added to each information page to play the television program and/or trailer associated with an information page when it is navigated to and comes to the front.

**Fig. 15** is a flow diagram of an exemplary method 1500 of multi-axis TV navigation. This exemplary method 1500 can be performed by a module or engine, such as the exemplary navigation engine 904 shown in Fig. 10. In the flow diagram, the operations are summarized in individual blocks. The operations

1 may be performed in hardware and/or as machine-readable instructions (software  
2 or firmware) that can be executed by a processor.

3 At block 1502, combinable navigation contexts for creating a list of  
4 television programs are selected.

5 At block 1504, the combinable navigation contexts are logically combined.

6 At block 1506, a database of television programming metadata is queried  
7 for television program identifiers associated with the combined navigation  
8 contexts.

9 At block 1508, television programs associated with the identifiers are  
10 presented for navigation.

## 11 12 CONCLUSION

13 It should be noted that the subject matter described above can be  
14 implemented in hardware, in software, or in both hardware and software. In  
15 certain implementations, the exemplary system, engine, and related methods may  
16 be described in the general context of computer-executable instructions, such as  
17 program modules, being executed by a computer. Generally, program modules  
18 include routines, programs, objects, components, data structures, etc. that perform  
19 particular tasks or implement particular abstract data types. The subject matter  
20 can also be practiced in distributed communications environments where tasks are  
21 performed over wireless communication by remote processing devices that are  
22 linked through a communications network. In a wireless network, program  
23 modules may be located in both local and remote communications device storage  
24 media including memory storage devices.

1       The foregoing discussion describes exemplary systems and methods for  
2 multi-axis television navigation. Although the subject matter has been described  
3 in language specific to structural features and/or methodological acts, it is to be  
4 understood that the subject matter defined in the appended claims is not  
5 necessarily limited to the specific features or acts described. Rather, the specific  
6 features and acts are disclosed as exemplary forms of implementing the claims.